

~\AppData\Roaming\Sublime Text\Packages\User\D.cpp

```
1  #include <bits/stdc++.h>
2  using namespace std;
3
4  #define INF INT_MAX
5  vector<int> costs;
6  vector<vector<int>> paths;
7
8  void printMatrix(vector<vector<int>>& matrix) {
9      int n = matrix.size();
10     for (int i = 0; i < n; i++) {
11         for (int j = 0; j < n; j++) {
12             if (matrix[i][j] >= 900) {
13                 cout << " x" << " ";
14             } else {
15                 cout << setw(3) << matrix[i][j] << " ";
16             }
17         }
18         cout << endl;
19     }
20     cout << endl;
21 }
22
23 int reduction(vector<vector<int>> &distanceMatrix) {
24     int n = distanceMatrix.size();
25     int nodeValue = 0;
26
27
28     for (int i = 0; i < n; i++) {
29         int min = INF;
30         for (int j = 0; j < n; j++) {
31             if (distanceMatrix[i][j] < min) {
32                 min = distanceMatrix[i][j];
33             }
34         }
35         if (min != INF) {
36             nodeValue += min;
37             for (int j = 0; j < n; j++) {
38                 if (distanceMatrix[i][j] != INF) {
39                     distanceMatrix[i][j] -= min;
40                 }
41             }
42         }
43     }
44
45
46
47
48
```

```

49
50     for (int j = 0; j < n; j++) {
51         int min = INF;
52         for (int i = 0; i < n; i++) {
53             if (distanceMatrix[i][j] < min) {
54                 min = distanceMatrix[i][j];
55             }
56         }
57         if (min != INF) {
58             nodeValue += min;
59             for (int i = 0; i < n; i++) {
60                 if (distanceMatrix[i][j] != INF) {
61                     distanceMatrix[i][j] -= min;
62                 }
63             }
64         }
65     }
66
67     return nodeValue;
68 }
69
70 vector<vector<int>>
71 updateMatrix(int ageValue, vector<vector<int>> &distanceMatrix, int from, int to,
72 vector<int> currentPath) {
73     vector<vector<int>> updatedMatrix = distanceMatrix;
74     int n = distanceMatrix.size();
75
76     updatedMatrix[to][from] = INF;
77     for (int i = 0; i < n; i++) {
78         updatedMatrix[i][to] = INF;
79     }
80     for (int j = 0; j < n; j++) {
81         updatedMatrix[from][j] = INF;
82     }
83
84     cout << "Changed Matrix after setting row and column to inf: [" << from + 1 << " -> " << to
+ 1 << "]" << endl;
85     printMatrix(updatedMatrix);
86
87     currentPath.push_back(to);
88     paths.push_back(currentPath);
89
90     int totalCost = reduction(updatedMatrix) + distanceMatrix[from][to] + ageValue;
91
92     costs.push_back(totalCost);
93
94     return updatedMatrix;
95 }
96

```

```
97 void baseMethod(vector<vector<int>> &distanceMatrix) {
98     int n = distanceMatrix.size();
99     vector<bool> visited(n, false);
100
101     int nodeValue = reduction(distanceMatrix);
102     cout << "Initial Node Value: " << nodeValue << endl;
103
104     vector<int> currentPath = {0};
105     int from = 0;
106     visited[from] = true;
107
108     for (int count = 1; count < n; count++) {
109         costs.clear();
110         paths.clear();
111
112         vector<vector<int>> bestMatrix;
113         int minCost = INF;
114
115         for (int i = 1; i < n; i++) {
116             if (!visited[i]) {
117                 vector<vector<int>>
118                     nextMatrix = updateMatrix(nodeValue, distanceMatrix, from, i, currentPath);
119
120                 if (costs.back() < minCost) {
121                     minCost = costs.back();
122                     bestMatrix = nextMatrix;
123                 }
124             }
125         }
126
127         for(auto it : costs){
128             cout << it << " ";
129         }
130         cout << endl;
131
132         distanceMatrix = bestMatrix;
133         nodeValue = minCost;
134
135
136         for (int i = 0; i < paths.size(); i++) {
137             if (costs[i] == minCost) {
138                 currentPath = paths[i];
139                 from = currentPath.back();
140                 break;
141             }
142         }
143         visited[from] = true;
144     }
145     cout << endl;
146     cout << "Final Path: ";
```

```
147     for (int i = 0; i < currentPath.size(); i++) {
148         cout << currentPath[i] + 1 << " -> ";
149     }
150     cout << 1 ;
151
152     cout << endl;
153     cout << "Minimum Cost for Travelling : " << costs[0] << endl;
154 }
155
156 int main() {
157     int n;
158     cin >> n;
159
160     vector<vector<int>> distanceMatrix(n, vector<int>(n));
161
162     for (int i = 0; i < n; i++) {
163         for (int j = 0; j < n; j++) {
164             cin >> distanceMatrix[i][j];
165         }
166     }
167
168     cout << "Initial Distance Matrix:" << endl;
169     printMatrix(distanceMatrix);
170
171     baseMethod(distanceMatrix);
172 }
173
```